PentaMath

OPERATOR'S MANUAL

Penta Systems International, Inc.
1511 Guilford Avenue, Baltimore, Maryland 21202

The material in this manual is proprietary and is for information only. We reserve the right to modify and/or delete any material and capabilities described herein.

TABLE OF CONTENTS

Introduction

PentaMath Description—Rules and General Description ..... Intro. 1

Section 1

Begin Display Equation 1–1
Begin In-Text Equation 1–2
End Equation 1–3

Section 2

Assign Space Band Value 2–1
Insert Vertical Space 2–2

Section 3

Superiors and Inferiors 3–1

Section 4

Equation Identification Number 4–1

Section 5

Division 5–1

Section 6

Limit Placement 6–1

Section 7

Radicals 7–1

Section 8

Vertical Mobility 8–1

Section 9

Coefficients 9–1

Section 10

Alignment Points 10–1

Section 11

Equation Breaking 11–1
Main Break 11–1
Secondary Break 11–1
End Breaking 11–2
Institute Breaking 11–2
Cancel One Line Breaking 11–3
Institute One Line Breaking 11–3

Section 12

Valid Text Commands in Equations 12–1

Section 13

Drawing Rules 13–1

Section 14

Equation Data Set-Up 14–1
System Standards 14–1
Equation Data Base 14–2
PentaMath Command Listing 14–5
Legal Point Sizes 14–6
Oversized Character Setup 15–1
Oversized Character Listing 15–4

Section 16

Equation Errors 16–1

Section 17

Examples of Difficult Math 17–0

Introduction

PentaMath Equation Program

The PentaMath equation program is quite versatile. It can create a variety of mathematical equations, completely composed and positioned, from a minimum of input commands. There are fundamental rules which must be observed. It is suggested that you read all of the instructions at least once before attempting to use this program.

All justifiers are converted to a fixed 5 unit space (controlled by sys. stds. item 74) unless changed by command ([as). The proper positioning of characters horizontally and vertically will not allow for normal variance in justifier values. Therefore, all justifiers are converted internally to a fixed space.

Rail commands within the equation program are always *multi* upper/lower rail, regardless of their usage in text.

Refer to the abbreviated alphabetical listing of commands for identification purposes, and to the material which follows for a more complete description and method of application. A few of the general instructions follow:

1. To call in the equation program you must first determine if the equation to be set is to appear within text ([di) or as a display (stand-alone) equation ([be) and then key the appropriate begin equation command. Be sure to cancel all indentions before calling in the display equation program. Indentations *are* allowed in text equations.
2. Any measure, point size or leading that is in effect while keying equations *must* be defined using a true combination command—it cannot be a measure, point size, or leading defined by a variable. Kern pairs ([ak) must be cancelled ([xk) before calling the equation program, and re-instated, if wanted, after the equation. Tracking ([tr1, etc.) and increase or decrease character width ([iw, [dw) are automatically suspended by the equation program call and re-instated by end equation.
3. All Autosort Commands are valid in the PentaMath equation program (except those requesting accents). It is essential that autosorts be used for accessing characters which are considered equation break points. The autosort program inserts information with these characters to be used later for purposes of breaking equations within the measure being used. (Necessary accents are inserted by keyboarding for direct access instead of using autosorts or creating autosorts for these accents which do not use the accent program.)
4. If you are driving a VIP as your output device, special Mergenthaler mathematical italic fonts (PM–1) must be used to get the proper horizontal character fit of italic characters.

Introduction—Continued

5. Elements of equations are generally keyed in the same sequence as they would normally be read. For example, in fractions, the numerator is always keyed before the denominator. The division line is automatically generated by the program at its proper position and length. In general, equations are keyed left to right and top to bottom.

Example:

Sample Input:

```
[be[nuA[deB[ed[ee
```

Sample Output: (on 27 picas)

$$\frac{A}{B}$$

6. Superior characters and superior (or inferior) to superiors are all keyed before the inferior characters. The program will position them one above the other, in the proper size and sequence.

Example:

Sample Input:

```
[beA[esa[rb[eib[rb[ee
```

Sample Output: (on 27 picas)

$$A_b^a$$

First level superiors and inferiors are set in text superior size. The next level of superiors and inferiors are automatically coded for a smaller size until the minimum size of your phototypesetter is reached. At this point, the software continues to use this point size and moves the character up for superiors or down for inferiors.

Example:

Sample Input:

```
[beA[esa[esb[esc[esd[ese[rb[eia[eib[eic[eid[eie[rb[ee
```

Sample Output: (on 27 picas)

Introduction—Continued

Superiors and inferiors are automatically output from your superior and inferior italic fonts or roman fonts as specified in your Equation Data Base, negating the need to consequently issue change font commands normally associated with equations.

7. Most commands in the equation program are similiar to Autosort Commands in the way they change fonts. Those which return you to the baseline will also return you to the font used for the previous non-autosort baseline character—superior, inferior, oversized characters, and over/under commands.
8. The equation program must be terminated at the end of each equation by a [ee and recalled for the next equation. All display equations will be centered, unless a different quadding command is keyed immediately preceding the [ee.
9. The equation program is currently capable of setting in all valid point sizes up to a baseline size of 18 point.
10. If an equation is too long to fit the measure, the program will break the equation if an acceptable point is found in the Autosort program deck. PentaMath is aware that different break points are used for text equations and display equations, and handles these differences as described in Section 1–1 and 1–2.
11. Oversize symbols such as Summation, Product and Integral signs will be set oversize whenever accessed by the mnemonics in the list of Oversize Characters (Section 15–4).

Normal size will be obtained if requested by autosort commands.

12. Formats may not be used in PentaMath the same way they are used in text. The restrictions placed on the use of formats in equations are that the define format command cannot be used within equations, and all commands used in format must be valid to the equation program.
13. Variables *are not supported* in PentaMath due to the complexity of the program. Variables *are* supported in the same tape as long as they are not used within the actual equation (between [be and [ee).

Command Name: Begin Display (stand-alone) Equation **Command No. 88**

Input Code:
[be

Cancel Codes:
[ee
[te

Purpose:

This command activates software designed to handle equations which are to be set in position, yet apart from the preceding and following text. It must be issued at the beginning of each stand-alone equation.

Note: This command cannot be used in tabular.

This software is designed to:

Break equations within the current measure. Breaks are performed *before* the software-selected math symbol if keyed as an auto-sort.

Equations are broken according to the following:

1. Single line equations are automatically centered unless a quadding command is immediately specified before the end equation command.
2. If an equation makes two lines, the first line is flushed left on the current remaining measure, and the second line is flushed right on the current remaining measure unless cancel one-line breaking is in effect.
3. If the equation must be broken more than once, the system attempts to align the relational (i.e. $=, \leq, \geq$) symbols and center the overall structure. Further breaking will be explained in section 11.

Most text commands are not allowed when the equation program is activated, but are valid after the end equation command. However, change face, quad left, quad right, change point size, quad center, all formatting commands except define format, are valid in stand-alone equations ([be).

Caution:

1. If an equation number is desired, the equation number command must immediately follow the [be, and may only be preceded by a change face command.

2. An equation CANNOT be quadded in the same direction as an equation number.

Command Name: **Begin In-Text Equation** **Command No. 54**

Input Code:
[di

Cancel Codes:
[ee
[te

Purpose:

This command activates software designed to handle equations appearing within a paragraph (text). This software is designed to include the following features:

1. Break equations within the current measure. Breaks are performed *after* the appropriate software selected math symbol, if keyed as an autosort.
2. If the length of the equation is such that the runover portion will not fit within the measure, the runover lines are centered in the current measure until the remaining portion will fit within a text line.

Selection of this command indicates that the following equation is to be set within text or tabular. Any quadding commands inserted to actually quad the equation are illegal. An error message will be displayed on the computer console and the tape will be suspended. Quadding commands may be used following [ee.

Any "begin command" code followed by letters not within the Equation Command section will result in an error message and the tape is suspended.

Most text commands are not allowed when the equation program is activated, but are valid after the end equation command. However, change face, change point size, and formatting commands are allowed in text equations ([di). Leading is adjusted to accommodate superior and inferior.

Caution:
No [el__] command may be issued before a [di command. Any leading here must be issued as [ol__][ep.

Command Name: **End Equation** **Command No. 129**

Input Code:
[ee

Cancel Codes:
[te

Purpose:

This de-activates the equation program and activates PentaWare. If you are setting a display equation, this command will center it in the current measure.

All PentaWare commands are now valid. All equation commands are invalid. You are automatically returned to the typeface, point size and leading you were in when you requested PentaMath. Remember to re-establish any kerns, hangs, etc. that you want for text setting.

For purposes of typeface—remember: If your last character before entering PentaMath was an autosort, you would return to whatever face was in effect prior to keying the autosort.

Caution:

At the present time there are two equation commands not cancelled by [ee. These are Assign Space Band Value (See section 2–1) and Cancel Single Break (See section 11–3).

Command Name: Assign Space Band Value **Command No. 37**

Input Code:
[as

Cancel Codes:
Another [as
[te

Purpose:
The equation program automatically converts all spacebands to the fixed value specified in system standards item number 74.

This command allows you to change the value within the equation mode. The relative units are in relation to the point size. Multiple Assign Space Band value commands may be used within the equation. This command is specified as a 2 digit field.

Example:

(base of 54): [as54 = 54 units
[as09 = 9 units

Sample Input:

```
[be[as541 + 2 + 3 = 6[ee
```

Sample Output: (on 27 picas)

1 + 2 + 3 = 6

Caution:
This command stays in effect until a tape end ([te).

Command Name: Insert Vertical Space **Equation Command No. 30**

Input Code:
[iv

Cancel Codes:
Another [iv
[ee
[te

Purpose:
This command automatically inserts vertical space between every line of a stand-alone equation, whether broken by the keyboard operator or the equation program. This command is stated in ½ pt. increments and *may not* be used in in-text equations.

Example:

[iv12 = 6 pts.
[iv06 = 3 pts.

Sample Input:

```
[be[iv12] + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 - 9 - 8 - 7 - 6 - 5 -
4 - 3 - 2 - 1 = 0[ee
```

Sample Output: (on 27 picas)

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 - 9$$
$$- 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 = 0$$

Function: Superiors and Inferiors

Command Name:	**Superior**	**Equation Command No. 1**
Command Name:	**Inferior**	**Equation Command No. 2**
Command Name:	**Return to Baseline**	**Equation Command No. 4**
Command Name:	**Down Base (Return to Sub-base)**	**Equation Command No. 3**

Input Code:	**Cancel Codes:**
[es	[rb
[ei	[rb
[rb	None
[db	[rb
All above	[ee
All above	[te

Purpose:

The [es and the [ei commands select superior and inferior fonts, as described in the Equation Data base, unless another face is designated by a change face or rail command (after keying [es or [ei). Superiors to full size must be keyed before inferiors if both are present. Keying in this order will automatically stack the superiors and inferiors. A [rb command must be issued before beginning inferior to full size whenever superiors are present. If staggered superiors and inferiors are desired, autosorts or change face should be used.

Example:

Sample Input:

```
[beA[esab[rb[eicdef[rb    A;ya[esbcdef[rb[ee
```

Sample Output: (on 27 picas)

$$A_{cdef}^{ab} \qquad A_{a}^{bcdef}$$

Purpose:

Multiple [es commands may be used to get superior to superior ([ei for inferior to inferior, etc.). The [db command is used to return the program to the baseline of the last lower order of superior or the last higher order of inferior keyed. Before any full size character can be keyed, the [rb command must be keyed to restore the program to full size characters on the normal baseline. A [rb command will immediately return to the normal baseline without the need of intermediate [db commands.

Example:

Sample Input:

```
[beA[esa[esb[esc[dbb[esc[rb[eia[eib[eic[dbb[eic[rbA[ee
```

Sample Output: (on 27 picas)

$$A_{a_{b,b_c}}^{a^{b^{c^{bc}}}}A$$

Function: **Superiors and Inferiors (Cont.)**

Purpose:
The [es and [ei commands generate the necessary coding for superiors and inferiors in relation to the size of the last character accessed, and will align them according to the height of that character.

Example:

Sample Input:

```
[b2[nuA[deB[ed[t2[esa[rb[eib[rb[ee
```

Sample Output: (on 27 picas)

$$\left(\frac{A}{B}\right)_{b}^{a}$$

Purpose:
When [es and [ei commands are used to define the limits of an integral sign they will position higher and lower than for a normal text size character as above. The lower (inferior) characters will be kerned or backed under the integral sign.

Example:

Sample Input:

```
[be[ig[esa[rb[eib[rb[ee
```

Sample Output: (on 27 picas)

$$\int_{b}^{a}$$

Caution:
Commands must be completed—if a superior or inferior command is used, it must be followed by a return to base command before ending the equation. The same holds true for second order superiors or inferiors, a return to base command must be used.

Command Name: **Equation Identification Number** **Equation Command No. 19**

Input Code:
[en

Cancel Codes:
[ql
[qr
[ee
[te

Purpose:

This command is designed for those equations needing "equation identification numbers". Its employment demands two prerequisites:

1. It must immediately follow a [be command. If a [di command was keyed instead of a [be, it will create an error message.
2. It must be followed by a quadding command in order for the software to determine left or right placement.

Its necessity of being keyed directly after the [be command is intentional. The total width of the equation identification is necessaray in order for PentaMath to properly compute its structuring and breaking.

A pica is added to the accumulated widths of the number keyed to insure that the I.D. does not touch the typeset equation.

If the equation's length is such that the equation will fit on the line but the I.D. will not, the equation is broken. Its left or right placement will be specified by the quadding command at the end of the equation identification. If quad left was specified, the equation number will be on the first line, if quad right it will be on the second line of the broken equation. If the broken equation makes more than two lines the number will appear quadded correctly and it will be vertically centered on the number of lines.

Example:

Sample Input:

```
[be[en(1)[qrAB = CD[ee
```

Sample Output: (on 27 picas)

$$AB = CD \qquad (1)$$

Function: Equation Identification Number (Cont.)

Note: The example below uses alignment point mnemonics ([lp and [le) which are explained later in this manual.

Example:

Sample Input:

```
[be[en(2)[qrAB = [lpCD[le[lpEF[le[lpGH[le[ee
```

Sample Output: (on 27 picas)

$$\begin{aligned} AB &= CD \\ &\quad EF \qquad (2) \\ &\quad GH \end{aligned}$$

Purpose:

It is also possible to have multi-equation numbers for doing alignment of multiple equations. When multiple equation numbers are used they are keyed one after the other with only the last one requiring the quadding function. If you key five equation numbers and only four equation lines are set, the fifth number will be discarded.

Example:

Sample Input:

```
[be[en(1)[en(2)[en[en(3)[qrA [lp= B[le[lp= C[le[lp= D[le[lp=
E[le[ee
```

Sample Output: (on 27 picas)

$$\begin{aligned} A &= B \qquad (1) \\ &= C \qquad (2) \\ &= D \\ &= E \qquad (3) \end{aligned}$$

Caution:

1. **Only one quadding command in multiple or forced equation numbers is allowed.**
2. **Failure to place a quadding command at the end of the equation identification will result in an error.**

Function: Fractions—Division

Command Name:	**Numerator**	**Equation Command No. 6**
Command Name:	**Denominator**	**Equation Command No. 5**
Command Name:	**End Division**	**Equation Command No. 7**

Input Code:	**Cancel Codes:**
[nu	[de
[de	[ed
[ed	None
All above	[ee
All above	[te

Purpose:

A fraction consists of three basic parts as illustrated below. In the PentaMath program, if you properly describe the placement of the numerator and the denominator, the program will automatically create the divisional rule to the proper length.

$$\text{division rule} \quad \frac{X + Y}{Z} \quad \begin{matrix}\text{numerator} \\ \text{denominator}\end{matrix}$$

Since division and fractions are read from the top down, the numerator command ([nu) is used not only to denote the numerator but also to denote the beginning of the fraction. The denominator command terminates the numerator and begins the denominator. The fraction is terminated by the end division ([ed) command. To demonstrate the use of these commands, the fraction shown above would be keyed:

Example:

Sample Input:

```
[be[nuX + Y[deZ[ed[ee
```

Sample Output: (on 27 picas)

$$\frac{X + Y}{Z}$$

Purpose:

In more complicated divisions, the numerator and/or the denominator may also contain multiple levels of division, oversized characters, stacked superior/inferiors, etc. In the following examples we will cover how to set some complicated divisions.

The first example will cover division within division. To make the example easier to comprehend, the secondary divisions are grouped within brackets.

Note: In the following descriptions the "&" equals a tape feed (unit space).

Function: Fractions—Division (Cont.)

Purpose:

Example:

Sample Input:

```
[beA = [nu&[nuA + B[de2[ed&[deC + DE[ed
       + [nu&[nuI + K[de2 × 2[ed&[de&[nu3X[deH + S[ed&[ed
       + [nu2[de&[nu2M[de2X + Y[ed&[ed[ee
```

Sample Output: (on 27 picas)

$$A = \frac{\left[\frac{A + B}{2}\right]}{C + DE} + \frac{\left[\frac{I + K}{2 \times 2}\right]}{\left[\frac{3X}{H + S}\right]} + \frac{2}{\left[\frac{2M}{2X + Y}\right]}$$

In order to further understand the above division, it is rewritten below treating the bracketed areas as single terms.

$$[beA = [nu \left[\frac{A + B}{2}\right] [deC + DE[ed + [nu \left[\frac{I + K}{2 \times 2}\right] [de \left[\frac{3X}{H + S}\right] [ed + [nu2[de \frac{2M}{3X + Y} [ed[ee$$

The description to the computer can be completed by encoding the divisions enclosed in brackets as usual, except for the fact that subdivisions are separated by a tape feed (unit space).

An important aspect to remember when doing subdivisions is to make sure your coding "equals out"—meaning that if you have 3 numerators, then you must also have 3 denominators and 3 end divisions. Following are some further examples of division within division. Notice that the program automatically aligns the main divisional rule.

Example:

Sample Input:

```
[be[nul + [nul[de2[ed&[de3[ed    [nu3[del + m [nul[de2[ed&[ed[ee
```

Sample Output: (on 27 picas)

$$\frac{1 + \frac{1}{2}}{3} \quad \frac{3}{1 + m\frac{1}{2}}$$

Function: Limit Placement

Command Name:	**Mark**	**Equation Command No. 11**
Command Name:	**Begin Over**	**Equation Command No. 8**
Command Name:	**Begin under (Below)**	**Equation Command No. 10**
Command Name:	**End Over/Under**	**Equation Command No. 9**

Input Code:	**Cancel Codes:**
[mk	None
[ov	[bl
[bl	None
[eo	None
All above	[ee
All above	[te

Purpose:

These commands are used chiefly to place limits on Summation and Product symbols, but can also be used at any time within the equation program to place superior and inferior characters above or below phrases or terms. The Mark command is essential to denote the beginning of the information you wish to place something over and/or under.

Example:

Sample Input:

```
[be[mkLIM[blna[eo[ee
```

Sample Output: (on 27 picas)

$$\underset{na}{\text{LIM}}$$

Purpose:

A mark code is automatically generated by calling for an oversize character.

Example:

Sample Input:

```
[be[ns[ovn[blij[eo[ee
```

Sample Output: (on 27 picas)

$$\sum_{ij}^{n}$$

Function: **Limit Placement (Cont.)**

Purpose:
This automatic mark can be overriden by typing a mark code.

Example:

Sample Input:
```
[be[mk[ns[ns[ovn[blij[eo[ee
```

Sample Output: (on 27 picas)

This is necessary when the limits center over two or more oversize characters. The end of the unit you wish to place something over or under is determined by typing the appropriate Over and/or Under command. If both, an Over term and an Under term are required, the over term must be keyed first. The end of the last term to be placed Over or Under is denoted by the End Over/Under command, i.e., [eo.

Purpose:
Multiple over/under can be accomplished as follows.

Example:

Sample Input:
```
[be[mkLIM[ov[ov[ovM[blL[blK[blnn[blmm[blzz[eo[ee
```

Sample Output: (on 27 picas)

All oversized characters for which there is a special command will automatically generate a Mark command. If the character has no Over/or Under command following it, the Mark command is ignored.

Mark codes are *always* necessary *except* on oversized characters.

Function: Radicals (Square Root)

Command Name:	Begin Radical Rule	Equation Command No. 12
Command Name:	End Radical Rule	Equation Command No. 13
Command Name:	Radical symbol—one line high	Equation Command No.
Command Name:	Radical symbol—two lines high	Equation Command No.

Input Code:	Cancel Codes:
[br	[er
[er	None
[r1	None
[r2	None
All above	[ee
All above	[te

Purpose:

These commands enable setting radical expressions. A radical expression consists of three basic parts as illustrated below. When keyboarded properly, PentaMath software will calculate the widths and leadings necessary to typeset the rule at the proper height and length.

These commands should not be used for overscoring words in text because the precise height of the radical is used to determine the height at which the rule is placed.

Radical rule
Radical Symbol $\sqrt{A + 4}$ Radicand

Example:

Sample Input:

```
[be[r1[brA[er[ee
```

Sample Output: (on 27 picas)

$\sqrt{A}$

Function: Radicals (Cont.)

Example:

Sample Input:

```
[be[r2[br[nuA[deB[ed + [nuD[deE[ed[er[ee
```

Sample Output: (on 27 picas)

$$\sqrt{\frac{A}{B} + \frac{D}{E}}$$

Example:

Sample Input:

```
[be[r2[br[nu[rl[brA + B[er[de[rl[brC[er[ed[er[ee
```

Sample Output: (on 27 picas)

$$\sqrt{\frac{\sqrt{A + B}}{\sqrt{C}}}$$

Function: Vertical Mobility:

Command Name:	**Move Up**	**Equation Command No. 18**
Command Name:	**Move Down**	**Equation Command No. 17**

Input Code:	**Cancel Codes:**
[mu	[md
[md	[mu
Both above	[ee
Both above	[te

Purpose:

These commands give keyboard operators the ability to move up or down any time and in any one half point amount they desire. Each command is followed by a two digit field which specifies the vertical movement in half points.

Example:

[mu12 = 6 pts.
[mu06 = 3 pts.

One possible usage of this command would be to prevent second order superiors and inferiors overlapping each other as in the following example.

Example:

Sample Input:

```
[beH[esA[eig[rb[eiA[esd[rb  H[mu06[esA[eig[rb[md06
[md06[eiA[esd[rb[ee
```

Sample Output: (on 27 picas)

$H_{A^{a}}^{A^{s}}$ $H_{A^{d}}^{A^{s}}$

Function: Coefficients (Matrices)

Command Name:	**Number of Coefficients**	**Equation Command No. 15**
Command Name:	**Finish Term (same as [qc)**	**Equation Command No. 16**
Command Name:	**Quad Left**	**Equation Command No. 33**
Command Name:	**Quad Right**	**Equation Command No. 34**
Command Name:	**Quad Center**	**Equation Command No. 35**

Input Code:	**Cancel Codes:**
[nc	Fulfillment of count
[ft	None
[ql	None
[qr	None
[qc	None
All above	[ee
All above	[te

Note: Coefficients are full-size terms stacked one above the other.

Purpose:

To begin setting coefficients, the first command must be [nc (Number of Coefficient terms) followed by a single digit, which is the number of lines within the coefficient. The terms within the coefficient are typed in sequence from top to bottom, and each term must be terminated by [ft or a quadding command. After the number of terms defined in the [nc command has been satisfied, the program will center or quad all the terms on the longest one.

Example:

Sample Input:

```
[beA + [o2[nc2sin[qlcos[ql[e2[ee
```

Sample Output: (on 27 picas)

$$A + \begin{bmatrix} \sin \\ \cos \end{bmatrix}$$

Function: Coefficients (Cont.)

Example:

Sample Input:

```
[be[nc3l  0  l[qlO  l  O[qlO  l  O[ql[ee
```

Sample Output: (on 27 picas)

```
1  0  1
0  1  0
0  1  0
```

Caution:

1. **Division and begin radical commands are not supported within coefficient command.**
2. **Maximum of 9 terms within coefficients.**

Function: Alignment Points

Command Name: Lineup Point — Equation Command No. 23
Command Name: End of Equation Line — Equation Command No. 24

Input Code:	Cancel Codes:
[lp	[le
[le	None
Both above	[ee
Both above	[te

Purpose:

Quite often, when setting equations, it is necessary that different parts of the equation be aligned. PentaMath allows for ten alignment points per typeset line and a maximum of ten lines of equation. The [lp command is used to denote the points of alignment within each line and [le is used to end each line. Each line of the equation must contain the same number of alignment points. The following examples will better demonstrate the usage of these commands.

Example:

Sample Input:

```
[bex² + 4x [lp= (x + 2)[le[lp= (5 + 3)[le[lp= 11[le[ee
```

Sample Output: (on 27 picas)

$$\begin{aligned} x^2 + 4x &= (x + 2) \\ &= (5 + 3) \\ &= 11 \end{aligned}$$

Function: Alignment Points (Cont.)

Purpose:

If, after the last [le, you insert a quadding command, the entire aligned structure is quadded in that direction.

If a single line within the structure contains no alignment points and is terminated with a quadding command, the line will be quadded in the designated direction without regard to the alignment of the other lines of the equation.

Example:

Sample Input:

```
[be[iv12F [lp= ma    [lpNEWTON[le
        m [lp= [num0[de[rl[srl - 2[er[ed    [lpEINSTEIN[le
        and[ql
        F + 1 [lp= [num0a[de[r2[srl + [nul[de2 + xd[ed[er
          [lpFORCE EQUATION[le[ee
```

Sample Output: (on 27 picas)

$$F = ma \qquad \text{NEWTON}$$

$$m = \frac{m_0}{\sqrt{1 - 2}} \qquad \text{EINSTEIN}$$

and

$$F + 1 = \frac{m_0 a}{\sqrt{1 + \frac{1}{2 + xd}}} \qquad \text{FORCE EQUATION}$$

Caution:

When using more than one alignment point in a line the first term will (in effect) quad right to the alignment point; subsequent terms will quad left.

Function: Equation Breaking

Command Name: Main Break **Equation Command No. 21**

Input Code:
[mb

Cancel Codes:
[ee
[te

Purpose:

This command is used to mark a break point in an equation. Multiple break points may be used in the same equation. This command functions differently within the stand-alone equation mode and the text equation mode.

In text, this command is treated as any other break point. No distinction is made regarding relational or conjunctive breaks.

In stand-alone, it will be treated as a relational break point if the equation is to be broken.

Autorsorts can be set up specifying mathematical functions (=, +, etc.) as relational or conjunctive break points. For example, you could set up your equal sign, greater than, less than, greater than or equal to, less than or equal to symbols, as relational break points and your plus sign, minus sign, times sign as conjunctive break points.

This command is used mostly as a 2nd pass editing command to change breaks that were done using breaking routines internally, or to assure a predetermined break if break is indeed needed as determined by the length of the equation.

Note: Will not break equation at this point if equation need not be broken.

Command Name: Secondary Break Point **Equation Command No. 22**

Input Code:
[sb

Cancel Codes:
[ee
[te

Purpose:

This command is used to mark a break point in an equation. Multiple break points may be used in the same equation. This command functions differently within the stand-alone equation mode and the text equation mode.

In text, it is treated as any other break point. No distinction is made regarding relational or conjunctive breaks.

In stand-alone, this command is treated as a conjunctive break point if the equation is to be broken.

As in the main break command, this command is used primarily as an editing tool.

Command Name: End Breaking **Equation Command No. 32**

Input Code:
[eb

Cancel Codes:
[ib
[ee
[te

Purpose:
This command allows the user to stop all program breaking within a specific equation. The computer does not recognize any break points after the "end breaking" command. If you wish to isolate an area you may issue an "end breaking" command. Then, after the area, issue an "institute breaking" command.

If you had a plus sign coded as a break point in the autosort file and your equation contained the following expression:

Example:

Sample Input:
```
[be[eb(a+b) [ib[ee
```

Sample Output: (on 27 picas)

(a + b)

Keying as above would prevent breaking on the plus sign.

Command Name: Institute Breaking **Equation Command No. 31**

Input Code:
[ib

Cancel Codes:
[eb
[ee
[te

Purpose:
This command allows the user to re-establish the breaking status of characters. Normally, this will be used after you have entered an "end breaking" command. This command may be used repetitiously within an equation. If no [eb was ever issued this command is not needed.

Command Name: Cancel One Line Breaking **Equation Command No. 25**

Input Code:
[co

Cancel Codes:
[ee
[te

Purpose:
This command is used to cause alignment of equal signs (multi break logic) and not quad left and right logic (single break logic).
The program recognizes the command and switches to the alignment of equal sign logic until the end of tape or a [io command is encountered.

Caution:
This command is not cancelled by end equation command. Only [io will cancel.

Command Name: Institute One Line Breaking **Equation Command No. 26**

Input Code:
[io

Cancel Codes:
[ee
[te

Purpose:
This command is used to re-establish the single line breaking logic within equations. This breaking logic takes the first part of the equation and quads it left and the second part and quads it right. If more than two lines are involved, the multi-breaking logic takes place. The computer will break single line and quad first part left and remainder right.

Function: Miscellaneous Text Commands

Command Name:	Get Job Format	**Equation Command No. 85**
Command Name:	Get Universal Format	**Equation Command No. 84**
Command Name:	Change Face	**Equation Command No. 36**
Command Name:	Change Point Size	**Equation Command No. 20**

Input Code:	**Cancel Codes:**
[j—]	[ee
[u—]	[te
[cf—]	
[cp—]	

Purpose:

These four text commands are usable in the PentaMath program. They function the same way as they do in text. However, the change face and change point size commands may have only *one* field each.

In text, the change face command may specify face, point size, and leading. In equations, it can only specify the face followed by an end command code.

In text, the change point size may specify point size and leading. In equations, it can only specify the point size followed by an end command code.

Function: Drawing Rules

Command Name:	**Underline Rule**	**Equation Command No. 38**
Command Name:	**Strike-Out Rule**	**Equation Command No. 39**
Command Name:	**User Position Rule**	**Equation Command No. 40**

Input Code:	**Cancel Codes:**
[lu	[er
[so	[er
[pr	[er
All above	[ee
All above	[te

Purpose:

The underlining-rule is used to obtain an underscore for one or more characters within the equation mode. This command has no fields and is terminated with an end-rule ([er) command.

Example:

Sample Input:

```
[be[lul + 2 + 3 = y[er[ee
```

Sample Output: (on 27 picas)

$$\underline{1 + 2 + 3 = y}$$

Function: Drawing Rules (Cont.)

Purpose:
The strike-out rule is used to obtain a rule through one or more characters within the equation mode. This command has no fields and is terminated with an end-rule ([er) command.

Example:

Sample Input:

```
[be[soa + b + c = m[er[ee
```

Sample Output: (on 27 picas)

~~a + b + c = m~~

Purpose:
The user position rule is used to obtain a rule in any position desired within the equation mode. This command has one field (height above the base-line stated in half-points) and is terminated with an end-rule ([er) command.

Example:

Sample Input:

```
[be[pr18A + B + C = D[er[ee
```

Sample Output: (on 27 picas)

$\overline{A + B + C = D}$

MATH DATA SET-UP

There are four sets of data used by the equation program. They are system standards (syss), equation data base (eqdb), oversize characters (osch) and autosorts (aupi). All four sets of data should be loaded before attempting to typeset equations. The following pages will explain how each of these data bases is to be filled out in order to use the math program.

SYSTEM STANDARDS

The first data base we will look at is the System Standards. Items 24, 25, 69, and 74 are used by the math program.

The math program is capable of automatically issuing a user command (see description of output module for information on user commands) before and after each equation. Item 24 is the number of the user command to be output before each equation and item 25 is the number of the user command to be output after each equation. If your output device doesn't require a user command before and/or after each equation, put a -1 in the appropriate item(s)

When setting equations, it is necessary for the math program to calculate point sizes. This calculation is done in hundredths of a point and must be corrected to a valid point size which the output device can perform. Item 69 of System Standards is used to make this correction by the following method:

$$\text{Usable point size} = \left(\frac{\text{calculated point size (in hundredths)}}{\text{item 69}}\right) \times \text{item 69}$$

If the output device is capable of setting only full point sizes, item 69 should be 100. If it is capable of setting in tenth of a point increments, item 69 should be 10.

In math, all variable spaces are converted to fixed space. Item 74 determines the number of relative units each variable space is to be converted to.

```
                                EQUATION DATA BASE TABLE


BASE SUPERIOR FONT # --    12 (SERIF)
BASE INFERIOR FONT # --    16 (SERIF)
BASE SUPERIOR FONT # --    12 (SANS SERIF)
BASE INFERIOR FONT # --    16 (SANS SERIF)
                           FONT  CHAR.  SHOVE  WIDTH  HEIGHT

          PRIMARY RULE-----  -1    M      3     54      6
          SECONDARY RULE---  -1    L      3     54      6
          RADICAL RULE-----  -1    L      3     54     16
          UNDERSCORE-------  -1    L      3     54     -4
          STRIKE OUT-------  -1    M      3     54      6
          USER POSITION----  -1    P      3     27      6

                                          =====COMMANDS=====

  1.ES   2.EI   3.DB   4.RB   5.DE   6.NU   7.ED   8.OV   9.EO  10.E
 11.MK  12.BR  13.ER  14.EE  15.NC  16.FT  17.MD  18.MU  19.EN  20.C
 21.MB  22.SB  23.LP  24.LE  25.CO  26.IO  27.SM  28.NM  29.FY  30.I
 31.IB  32.EB  33.QL  34.QR  35.QC  36.CF  37.AS  38.LU  39.SO  40.F

POINT SIZES:
 -1   5 72
```

EQUATION DATA BASE

The Equation Data Base contains the fonts to be used for superiors and inferiors, the description of how to draw rules, the command mnemonics and the valid point sizes.

Following is an example of the equation data base broken down into 4 sections. The material in boldface represents the fields which must be filled in.

Section A

BASE SUPERIOR FONT # – **11** (SERIF)
BASE INFERIOR FONT # – **15** (SERIF)
BASE SUPERIOR FONT # – **11** (SAN SERIF)
BASE INFERIOR FONT # – **15** (SAN SERIF)

If your system uses a mix, then for the base superior and inferior fonts you put in the actual font numbers the math program is to access whenever a superior or inferior command is encountered in an equation.

On systems which use families, the number which is inserted is the member of the family that is superior and the member which is inferior. (NOTE: This means that the superiors and inferiors must occupy the same positions in all families).

The system has the ability to select either serif or san serif superiors and inferiors depending on the typeface in use when the superior or inferior command is encountered. If you elect to use only one set of superiors and inferiors then the serif lines are filled in and the san serif lines are both zero.

Section B

		FONT	CHAR.	SHOVE	WIDTH	HEIGHT
PRIMARY RULE	-	-1	M	3	54	6
SECONDARY RULE	-	-1	L	3	54	6
RADICAL RULE	-	-1	L	3	54	16
UNDERSCORE	-	-1	L	3	54	-4
STRIKE OUT	-	-1	M	3	54	6
USER POSITION	-	-1	L	3	54	6

Field 1 (FONT)

May contain one of the following three values:

-1 Indicates that the ruling character is available in all fonts.
0 Output ruling command of phototypesetter to draw rule.
— Number of font which contains ruling character.

Field 2 (CHAR.)

If the first field (FONT) is not zero, then this field is the character to be used to draw the rule.

If the first field *is* zero, then this field is also zero.

Field 3 (SHOVE)

If the first two fields (FONT and CHAR.) are not zero, then field 3 is the keyboard status of the ruling character.

0 = Unshove
1 = Shove
2 = Secondary mode unshove
3 = Secondary mode shove

If the first two fields *are* zero, then this field is also zero.

Field 4 (WIDTH)

If the first three fields (FONT, CHAR. and SHOVE) are not zero, then this field is the width of the ruling character in relative units.

If the first three fields *are* zero, then this field is the weight of the rule in tenths of a point.

Field 5 (HEIGHT)

If the first field (FONT) is positive, then this is the distance (in half points) from the base to the ruling character when exposed in ten point type.

If the first field is zero *or* negative, then this is the base deflection (in half points) to be used to properly align the rule in a 10 point equation. Normal values are 5 or 6 for the divisional rules and 16 or 18 for the radical rule.

Section C

====COMMANDS====

1.ES	2.EI	3.DB	4.RB	5.DE	6.NU	7.ED	8.OV	9.EO	10.BL
11.MK	12.BR	13.ER	14.EE	15.NC	16.FT	17.MD	18.MU	19.EN	20.CP
21.MB	22.SB	23.LP	24.LE	25.CO	26.IO	27.SM	28.NM	29.FY	30.IV
31.IB	32.EB	33.QL	34.QR	35.QC	36.CF	37.AS	38.LU	39.SO	40.PR

A two character alphabetic mnemonic must be loaded for all math commands. A mnemonic must be loaded for *all* math commands even if one elects not to use the command. The mnemonics selected for equation commands may not be the same as in text or pagination, although the equation commands secondary mode, normal mode, change family, quad left, quad center, quad right, change face and change pointsize may have the same mnemonic as they do in text. The list on the next page gives the equation command number, the suggested mnemonic and the command name or function for all math commands.

Section D

POINT SIZES
−1 5 72

The final part of the equation data base which must be filled in, is the point sizes available on the phototypesetter.

For machines such as the VIP, all point sizes are entered in ascending order.

For machines which are capable of accessing all point sizes in a given range, the first field is −1, the second field is the minimum point size, and the third field is the maximum point size.

OVERSIZED CHARACTERS

The oversize character data base is next to be loaded. For each oversized character, ten data fields must be filled in to describe where to get the character, how to set the character, and what the physical size of the character is.

The data is loaded for how the character is to be handled in a ten point equation, and all other sizes are scaled from the ten point version.

Fifty oversized characters can be defined by the user. Following is a list of 25 characters that are supplied with the system.

OVERSIZE CHARACTER FILE
25 CHARACTERS TOTAL

MNEMONIC	FONT #	SHOVE	CHAR.	POINT SIZE	BASE ADJUST	WIDTH	HEIGHT	% ABOVE BASE	KERN ON I
NS	5	0	P	24	-2	24	10	80	0
IG	5	0	M	24	0	24	20	80	3
R1	5	0	I	10	-2	10	9	100	0
02	5	0	C	30	-4	30	27	75	0
E2	5	0	D	30	-4	30	27	75	0
03	5	0	2	30	-13	30	36	65	0
E3	5	0	3	30	-13	30	36	65	0
B2	5	0	A	30	-4	30	27	75	0
T2	5	0	B	30	-4	30	27	75	0
B3	5	0	&	32	-12	32	36	65	0
T3	5	0	1	32	-12	32	36	65	0
L2	5	0	E	30	-4	30	27	75	0
F2	5	0	F	30	-4	30	27	75	0
L3	5	0	4	31	-14	31	36	65	0
F3	5	0	5	31	-14	31	36	65	0
M2	5	0	J	30	-4	30	24	80	0
M3	5	0	9	30	0	30	36	80	0
V2	5	0	G	30	-4	30	27	75	0
V3	5	0	6	30	-14	30	36	80	0
D2	5	0	H	30	-4	30	27	75	0
D3	5	0	7	30	0	30	36	80	0
BP	5	0	0	16	-4	16	8	80	0
R2	5	0	8	30	-12	34	36	65	
QB	2	0	?	10	0	1	1	100	
DV	2	0	!	12	-3	0	11	90	

The following pages explain the above printout.

OVERSIZED CHARACTERS (Cont.)

Field 1 (MNEMONIC)

This is the two character mnemonic that will be used to identify the oversized character. The first field of the mnemonic must be alphabetic, and the second may be either alphabetic or numeric.

The mnemonic *cannot* be the same as any mnemonic in text, math, or pagination.

Field 2 (FONT #)

The PentaWare font number of the font which contains the oversized character.

Field 3 (SHOVE)

The PentaWare shove status of the character to be output.

0 = Unshove
1 = Shove
2 = Secondary mode unshove
3 = Secondary mode shove

Field 4 (CHAR.)

The ASCII character which must be accessed to get the desired character.

Field 5 (SIZE)

The point size to be used for the oversized character when it is output in a ten point equation.

Field 6 (BASE ADJUST)

Baseline adjustment to be performed before outputting the character. This field is stated in half points. A positive value denotes upward movement, and a negative value denotes downward movement of the character.

Field 7 (WIDTH)

This is the set size to be used when the character is output in a ten point equation. This field is used only on CRT photo typesetters.

OVERSIZED CHARACTERS (Cont.)

Field 8 (HEIGHT)

This is the total height (in points) of the oversized character when exposed at the point size in field five. This field is used for rule generating and superior/inferior generation to obtain proper placement.

Field 9 (% ABOVE BASE)

This is the percentage of field 8 which is above the baseline. (This field is used for the same purpose as field 8 regarding rules and superiors/inferiors.)

Field 10 (KERNING ON INF.)

This is the number of relative units that an inferior following the oversized character is to be kerned. This field is used mainly to kern the lower limit on integrals.

EQUATION ERRORS

Number	Reason
1	Invalid command (see command table)
2	Error in the equation number
3	No mark for Over
4	Invalid coefficient count (maximum number is 9 or character other than a number uses coefficient count)
5	Quadding command within display in text
6	Paren. Expression or single term within alignment wider than measure
7	Invalid vertical displacement
8	Invalid autosort (see autosort table)
9	Invalid point size (minimum is 6, maximum is 24)
10	End equation within equation number
11	Not all lines contain the same number of alignment points
12	Computer unable to break equation
13	Equation with alignment points will not fit
14	Incorrect superior/inferior construction
15	End equation within superior/inferior
16	Error on oversize character
17	Error on rule
18	Invalid family command
19	End Over/Under without an Over or Under
20	Finish term without being in coefficients
21	Alignment points within in-text equation ([di)
22	Begin radical rule without end radical rule
23	Invalid extra space defined on run-over lines
24	End equation within division
25	Extraneous end division command
26	Unable to load font
27	Extraneous return to base
28	Unable to determine length of rules
29	Unable to load font for divisional rules
30	Invalid field on rule definition command (user position)

OVERSIZED CHARACTERS

SUGGESTED NUEMONIC			CHARACTER	OPTION REQUIRED
N	S	Σ	NOTATIONAL SUM	MATH
I	G	∫	INTREGRAL	MATH
O	B	‾	OVER BAR	MATH
B	P	Π	PRODUCT	MATH
O	2	⌈	OPEN 2-LINE BRACKET	MATH
E	2	⌉	END 2-LINE BRACKET	MATH
O	3	⎡	OPEN 3-LINE BRACKET	MATH
E	3	⎤	END 3-LINE BRACKET	MATH
B	2	(	OPEN 2-LINE PARENTHESIS	MATH
T	2	)	TERMINATE 2-LINE PARENTHESIS	MATH
B	3	⎛	OPEN 3-LINE PARENTHESIS	MATH
T	3	⎞	TERMINATE 3-LINE PARENTHESIS	MATH
R	1	√	RADICAL 1-LINE	MATH
R	2	√	RADICAL 2-LINE	MATH
L	2	{	LEFT 2-LINE BRACE	MATH
F	2	}	FINISHING 2-LINE BRACE	MATH
L	3	⎧	LEFT 3-LINE BRACE	MATH
F	3	⎫	FINISHING 3-LINE BRACE	MATH
M	2	/	SHILLING 2-LINE	MATH
M	3	/	SHILLING 3-LINE	MATH
V	2	|	VERTICAL 2-LINE	MATH
V	3	|	VERTICAL 3-LINE	MATH
D	2	‖	DOUBLE 2-LINE VERTICAL	MATH
D	3	‖	DOUBLE 3-LINE VERTICAL	MATH

PENTAMATH COMMANDS

COMMAND NUEMONIC		CMND. NO.	COMMAND FUNCTION	OPTION REQUIRED
E	S	1	EQUATION SUPERIOR	MATH
E	I	2	EQUATION INFERIOR	MATH
D	B	3	DOWN BASE	MATH
R	B	4	RETURN TO BASE	MATH
D	E	5	DENOMINATOR	MATH
N	U	6	NUMERATOR	MATH
E	D	7	END DIVISION	MATH
O	V	8	OVER	MATH
E	O	9	END OVER/UNDER	MATH
B	L	10	BELOW	MATH
M	K	11	MARK	MATH
B	R	12	BEGIN RADICAL	MATH
E	R	13	END RADICAL	MATH
E	E	14	END EQUATION	MATH
N	C	15	NUMBER OF COEFFICIENTS	MATH
F	T	16	FINISH TERM	MATH
M	D	17	MOVE DOWN	MATH
M	U	18	MOVE UP	MATH
E	N	19	EQUATION NUMBER	MATH
C	P	20	CHANGE POINT SIZE	MATH
M	B	21	MAIN BREAK POINT (RELATIONAL)	MATH
S	B	22	SECONDARY BREAK POINT (CONJUNCTIVE)	MATH
L	P	23	ALIGNMENT POINT	MATH
L	E	24	ALIGNMENT (LINE) END	MATH
C	O	25	CANCEL ONE LINE BREAKING	MATH
I	O	26	INSTITUTE ONE LINE BREAKING	MATH
S	M	27	SECONDARY MODE	MATH
N	M	28	NORMAL MODE	MATH
F	Y	29	FAMILY	MATH
I	V	30	INSERT VERTICAL SPACE	MATH
I	B	31	INSTITUTE BREAKING	MATH
E	B	32	END BREAKING	MATH
Q	L	33	QUAD LEFT	MATH
Q	R	34	QUAD RIGHT	MATH
Q	C	35	QUAD CENTER	MATH
C	F	36	CHANGE FONT	MATH
A	S	37	ASSIGN SPACEBAND SIZE	MATH
L	U	38	LINE-UNDER RULE	MATH
S	O	39	STRIKE OUT RULE	MATH
P	R	40	POSITIONABLE RULE AS DEFINED BY USER	MATH